Amendments to the Claims

The listing of claims will replace all prior versions, and listings of claims in the application.

1. (Previously Presented) A method of processing an interrupt verification support mechanism in a computer system comprising a processor and an input for external interrupts communicatively coupled to the processor, the method comprising:

processing at least one actual instruction in the processor;

comparing data content of a program counter with data content of an interrupt register to determine if the data content of the program counter matches the data content of the interrupt register; and

if an external interrupt request is received by the processor or if the data content of the program counter matches the data content of the interrupt register, replacing the actual instruction in an instruction fetch stage of the processor with an interrupt pseudo-instruction.

2. (Currently Amended) The method of claim 1, further comprising:

processing at least one actual instruction in the processor in an instruction pipeline, wherein instructions are processed concurrently by an instruction fetch stage, an instruction decode stage, an instruction issue stage, an instruction execute stage and a result write-back stage.

3. (Cancelled)

4. (Previously Presented) The method of claim 1, wherein the interrupt pseudo-instruction is created by a co-processor connected to the processor.

5. (Currently Amended) The method of claim 1, further comprising:

simultaneously processing a plurality of instructions in an instruction pipeline of the processor, wherein the instruction pipeline including several instruction stages.

6. (Previously Presented) The method of claim 1, further comprising:

storing information of a sort of interrupt to use in a register of the processor.

7. (Cancelled)

8. (Previously Presented) A processor configured to support interrupt verification, the processor comprising:

a program counter;

a fetch register configured to store at least one actual instruction;

an interrupt register configured to store data content associated with an instruction that is to be interrupted;

a comparator configured to compare the data content of the interrupt register with data content of the program counter to determine if the data content of the interrupt register matches the data content of the program counter; and

a multiplexer configured to replace the at least one actual instruction with an interrupt pseudo-instruction if an external interrupt request is received by the processor

or if the data content of the program counter matches the data content of the interrupt register.

9. (Currently Amended) The processor of claim 8 further comprising:

an instruction fetch that includes the fetch register, the program counter, and the interrupt register, wherein the fetch register [[being]] is coupled to a first input of the multiplexer.

10. (Previously Presented) The processor of claim 9, wherein a second input of the multiplexer is coupled to the interrupt pseudo-instruction.

11. (Previously Presented) The processor of claim 9, wherein the comparator creates a high level signal if the data content of the program counter matches the data content of the interrupt register.

12. (Previously Presented) The processor of claim 9, wherein an output of the comparator is connected to a first input of an or-operator, and a second input of the or-operator is connected to an interrupt controller to enable the or-operator to create a high level signal if a signal received from the interrupt controller differs from a signal received from the comparator.

13.14. (Cancelled)

15. (Previously Presented) The processor of claim 9, wherein an instruction coming from an output of the multiplexer is sequentially processed in an instruction pipeline of the processor.

16. (Previously Presented) The processor of claim 9, wherein an instruction pipeline of the processor includes an instruction fetch stage, an instruction decode stage, an instruction issue stage, an instruction execute stage and a result write-back stage.

17. (Previously Presented) The processor of claim 9, wherein the interrupt pseudo-instruction effects instruction state stages required by the interrupt pseudo-instruction.

18. (Previously Presented) The processor of claim 16, wherein if an interrupt request or an interrupt pseudo-instruction is received by the processor, the processor is configured to cancel an instruction that is in the instruction fetch stage and to reissue the cancelled instruction starting at the instruction fetch stage.

19. (Previously Presented) The processor of claim 16, wherein if an interrupt request or an interrupt pseudo-instruction is received by the processor, the processor is configured to cancel an instruction that is in any instruction stage and to reissue the cancelled instruction starting at the instruction fetch stage.

20. (Previously Presented) The processor of claim 8, wherein the interrupt pseudo-instruction is created by a co-processor connected to the processor.

21. (Previously Presented) The processor of claim 20, wherein the processor is a core decoder processor and the co-processor is a decoding accelerator configured to assist the core processor with a decoding function.

22. (Previously Presented) The processor of claim 20, wherein the processor is a reduced instruction set computer (RISC) processor.

23-24. (Cancelled)